Ex2 : Node & Express

- **Read more Node.js materials**
  - **https://medium.freecodecamp.org/the-definitive-node-js-handbook-6912378afc6e**
  - **Download:** https://nodejs.org/en/download/
- **Review ho to do modules and how to use NPM efficiently**
  - **Modules docs:** https://nodejs.org/docs/latest-v12.x/api/modules.html
  - **More about modules:** https://medium.com/better-programming/node-js-modules-basics-to-advanced-2464001229b6 **(extra read)**
  - **https://www.sitepoint.com/beginners-guide-node-package-manager/**
- **Express tutorials:**
  - **https://www.youtube.com/watch?v=L72fhGm1tfE**
  - **https://expressjs.com/en/starter/hello-world.html**
    - **Continue the tutorial using the "next" button in the bottom of the page**
  - **https://expressjs.com/en/guide/writing-middleware.html**
  - **https://expressjs.com/en/guide/using-middleware.html**
  - **Express API: https://expressjs.com/en/4x/api.html**
- **Learn how to fetch()**
  - **https://javascript.info/fetch**

Submit a zip file <yourID>_<firstName>_lastName_EX2.zip (e.g. '043462598_Ohad_Assulin_EX2.zip').

1. Build readWrite.js which gets two arguments “file to read from” and “file to write to”.
    a. It reads the first file and writes the opposite text (char by char) to the output file
    b. So
        i. >node readWrite a.txt b.txt
        ii. should read a.txt and write the reversed text to b.txt
2. Develop a **WebServer** using express.js
    a. Build a calc.js module that allows starting a web-based calculator app which supports the following routes:
        i. <POST> /start zerofiy the shared variable M (M = 0)
        ii. <POST> /calc/add/:num sets M+= :num . It returns the new M
        iii. <POST> /calc/sub/:num sets M -= :num. It returns the new M
        iv. <PUT> /calc/multiply/:num sets M=:num * M. it returns the new M
        v. <PUT> /calc/divide/:num sets M=M/:num. It returns the new M
        vi. <GET> /calc/M returns M
        vii. <POST> /calc/reset sets M=0 and returns 0
        viii. <DELETE> /calc/del delete the session
        ix. Upon unknown request. You should return 404 (HTTP Status)
        x. Upon requests that throw exceptions, you should return 500 (HTTP Status)
        xi. Develop a calcTest.js (stand-alone process) which tests all the calc.js functionality using node-fetch to make requests to your server. calcTest should print what it tests and what was the result (OK/FAILURE)
            1. https://www.npmjs.com/package/node-fetch
    b. myExpress.js **module** that registers the following endpoints:
        i. <GET> /calc.html return an HTML page that allow running calculation via HTTP request to the calc.js module on the **backend**
        ii. <GET> /readme.html return EX1’s readme
        iii. <GET> /test.html returns EX1’s test.html (that should work, including importing the js files needed)
    c. Create a webServer.js requires myExpress and calc and actually starts listen to the network

Notice:

- Ex2 should be done individually
- Prioritize using async/await where ever it’s possible

Last Submission date: 17/6/2021